

REMARKS

In accordance with the foregoing, claims 1, 3, 5-7, 10, and 11 have been amended. Claims 1, 3, 5-7, 10, 11, 14, 15, 17-25, and 28-38 are pending, with claims 1, 14, 23, 25, 32, and 35 being independent. No new matter is presented in this Amendment After Final Rejection.

Applicants' Statement of Substance of Interview

The Interview Summary mailed September 21, 2010, for the telephone interview conducted on September 14, 2010, between the Examiner, Primary Examiner Li B. Zhen, and the undersigned attorney, Randall S. Svihla, is acknowledged. The applicants' statement of the substance of the interview required by the Interview Summary and MPEP 713.04 is interspersed below with the remarks.

Claim Amendments and Entry of Amendment After Final Rejection

Independent claim 1 and claims 3, 5-7, 10, and 11 depending directly or indirectly therefrom have been amended to recite a "non-transitory computer-readable storage medium" to obviate a potential rejection under 35 USC 101 as covering a signal *per se* as suggested in the OG Notice entitled "Subject Matter Eligibility of Computer Readable Media" published on February 23, 2010, at 1351 OG 212. Since these amendments are being made based on USPTO guidelines, it is submitted that these amendments do not raise new issues that would require further consideration and/or search.

For at least the foregoing reasons, it is submitted that entry of this Amendment After Final Rejection is proper under 37 CFR 1.116 and MPEP 714.12 and 714.13.

Claim Rejections Under 35 USC 103

Rejection 1

Claims 1, 3, and 5-7 have been rejected under 35 USC 103(a) as being unpatentable over Kanazawa et al. (Kanazawa) (U.S. Patent No. 6,580,870) in view of Jones et al. (Jones)

and Lamkin et al. '021 (Lamkin '021) (U.S. Patent No. 7,448,021). This rejection is respectfully traversed.

Claim 1

It is submitted that Kanazawa, Jones, and Lamkin '021 do not disclose or suggest the following features of independent claim 1:

the control information comprises an application program interface (API) that generates a report signal used to identify a buffering state of the markup document; and

the report signal is used by the apparatus to verify whether the markup document has been successfully preloaded into the buffer, whether the markup document cannot be read due to an error, and whether the markup document is being read.

The Office states that "[a]lthough Kanazawa teaches the ability to identify the buffering state, it does not specifically teach that the identification is enabled by control information as claimed." However, the Office states that it considers these features to be taught by paragraphs [0066] and [0068] of Jones, and states that "[i]t would have been obvious to one of ordinary skill in the art at the time Applicant's invention was made to combine these teaches because Kanazawa teaches identifying the buffering state and Jones teaches a way to enable identification of the buffering state that can be used when implementing the disclosure of Kanazawa."

However, Kanazawa already discloses "control information providing functionality to enable the apparatus to identify buffering state information of the markup document to be preloaded into the apparatus" as recited in claim 1 because column 18, lines 4-7, of Kanazawa, which is part of column 18, lines 2-13, of Kanazawa relied on by the Office, states that "the DVD playback control program 116 will receive the URL addresses and check to see if the HTML files corresponding to the URLs have already been cached in the image display apparatus (steps S404 to S407)." Accordingly, there would have been no reason for one of ordinary skill to incorporate the "control information providing functionality" features allegedly taught in paragraphs [0066] and [0068] of Jones into Kanazawa's apparatus as proposed by the Office. Rather, it is submitted that the combination of Kanazawa and Jones proposed by the Office is based solely on an impermissible hindsight reconstruction of the invention arrived at by reading

the applicants' disclosure, which is prohibited in an obviousness rejection under 35 USC 103(a) pursuant to MPEP 2145(X)(A).

Furthermore, Kanazawa's DVD playback control program 116 does not generate a report signal that is used by Kanazawa's apparatus "to verify . . . whether the markup document cannot be read due to an error, and whether the markup document is being read" as recited in claim 1. Although the Office considers paragraphs [0066] and [0068] of Jones to teach these features, it is submitted there would have been no reason for one of ordinary skill in the art to incorporate these features of Jones into Kanazawa's apparatus because Kanazawa's apparatus does not require these features of Jones to operate, and the Office has not identified any deficiency in Kanazawa's apparatus that might be corrected by incorporating these features of Jones into Kanazawa's apparatus, or any advantage that might be provided by doing so. Rather, it is submitted that the combination of Kanazawa and Jones proposed by the Office is based solely on an impermissible hindsight reconstruction of the invention arrived at by reading the applicants' disclosure, which is prohibited in an obviousness rejection under 35 USC 103(a) pursuant to MPEP 2145(X)(A).

Furthermore, it is submitted that Jones does not disclose or suggest "an application program interface (API) that generates a report signal used to identify a buffering state of the markup document; and the report signal is used by the apparatus to verify . . . whether the markup document cannot be read due to an error, and whether the markup document is being read" as recited in claim 1 as alleged by the Office. The Office considers these features to be taught by paragraphs [0066] and [0068] of Jones, which state as follows:

[0066] FIG. 10 shows a process for in-app [*sic*; should be out-of-app] preloader execution, in accordance with aspects of the invention. The preloader may preload resources based on a static resource list that may be updated with resource lists associated with hints as described earlier. When the preloader determines that the application has launched, the preloader may suspend itself from executing. In doing so, the preloader may discard the resource (or chunk of a resource) which it was currently trying to load, finish loading the resource or chunk of data before suspension, or indicate how much of the resource or chunk it was able to preload before suspending. When the preloader resumes preloading resources, it may start with where it left off.

[0068] When the preloader resumes preloading resources, it may determine that one or more resources that it was going to preload were loaded by the application or an in-app preloader

(block 1060). The preloader may make this determination by checking a cache, memory, a file system I/O API, or some other location, to see if the file is already located local to the client. If so, the preloader may simply proceed to the next resource to load in the resource list (block 1050). The preloader may determine the next resource, too, was already preloaded by the application or an in-app preloader. As part of its procedure, the preloader may simply check to determine whether a resource in a resource list already exists locally. The term "locally" refers to the resource stored in a point that is more or less instantly accessible and of high bandwidth. For example, the resource may be stored in memory, hard drive, LAN server cache, proxy, and the like. The term locally is not restricted to residing only on the user's machine [sic]

The Office considers the file system I/O API referred to in paragraph [0068] (not paragraph [0066] as incorrectly stated by the Office in the Final Office Action of August 3, 2010) to correspond to the "application program interface (API)" recited in claim 1. However, the Office has not explained which elements in the above paragraphs of Jones it considers to correspond to the "report signal" and the "apparatus" recited in claim 1, such that the Office has not established a *prima facie* case of obviousness with respect to claim 1. The applicants also pointed this out on page 13 of the Amendment of May 14, 2010, but the Office did not respond to this statement in the Final Office Action of August 3, 2010.

During the telephone interview conducted on September 14, 2010, the attorney pointed this out to the Examiner, and the Examiner stated that he considers the client 1205 in FIG. 12 of Jones to correspond to the "apparatus" recited in claim 1. Also, the Examiner pointed out that paragraph [0068] of Jones states that "[w]hen the preloader resumes preloading resources, it may determine that one or more resources that it was going to preload were loaded by the application or an in-app preloader . . . by checking a cache, memory, a file system I/O API, or some other location, to see if the file is already located local to the client." The Examiner stated that in his opinion, the preloader would call the file system I/O API and the file system I/O API would return a value indicating whether the file is already located local to the client. The Examiner stated that he considers the return value from the file system I/O API to correspond to the "report signal" recited in claim 1.

Paragraph [0072], lines 20-24, of Jones states as follows:

That is, both an out-of-app preloader and an in-app preloader may call a file system I/O API to load a particular resource. If the file

system I/O API determines that the file is found locally, it may simply report to the calling preloader that the file has been loaded.

However, it is not seen where the "report signal" indicating whether the file or resource has been loaded that is generated by the file system I/O API is used by the "apparatus" in FIG. 12 of Jones "to verify . . . whether the markup document cannot be read due to an error, and whether the markup document is being read" as recited in claim 1 as alleged by the Office.

The Office states as follows:

The I/O API in Jones determines whether a file is already loaded. If the file is already loaded, the file has been successfully preloaded. If the file is not loaded, further information is used to determine the status of the file. When the I/O API indicates that a file is not located local to the client (§68 and 69) and the preloader discards the resource or chunk of a resource which it was currently trying to load due to the launching of the application (§66), the status of the file is that it cannot be read due to an error (the error being the interruption of the reading process due to the launching of the application).

However, it is submitted that Jones' preloader discarding the resource (or chunk of a resource) that it is currently trying to load when the preloader determines that the application has launched is the way Jones' preloader normally operates. It is not seen where Jones states that this is an error as alleged by the Office, or how one of ordinary skill in the art could reasonably conclude that this is an error as alleged by the Office. It is submitted that the Office's characterization of this as an error is based solely on the applicants' disclosure, such that the combination of Kanazawa and Jones proposed by the Office is based solely on an impermissible hindsight reconstruction of the invention arrived at by reading the applicants' disclosure, which is prohibited in an obviousness rejection under 35 USC 103(a) pursuant to MPEP 2145(X)(A).

Furthermore, the "report signal" indicating whether the file or resource has been loaded that is generated by the file system I/O API as described in paragraph [0072], lines 20-24, of Jones is generated after the preloader has already discarded the resource (or chunk of a resource) which it was currently trying to load when the preloader determined that the application had launched. It is not seen where Jones discloses that the "apparatus" in FIG. 12 of Jones uses this "report signal" to verify that the reading of the resource (or chunk of a resource) was interrupted due to the launching of the application, which the Office considers to be an "error" as recited in claim 1.

Furthermore, claim 1 recites "the report signal is used by the apparatus to verify . . . the markup document cannot be read due to an error." It is not seen where Jones discloses or suggests that any of the resources that are preloaded by the preloader "cannot be read" as recited in claim 1. Although the reading of a particular resource by the preloader may be interrupted by the launching of the application, the reading of that resource continues when the preloader resumes preloading resources. See paragraph [0066], lines 11-13, of Jones, which states that "[w]hen the preloader resumes preloading resources, it may start with where it left off."

Accordingly, for at least the foregoing reasons, it is submitted that paragraphs [0066] and [0068] of Jones do not disclose or suggest "an application program interface (API) that generates a report signal used to identify a buffering state of the markup document; and the report signal is used by the apparatus to verify . . . whether the markup document cannot be read due to an error" as recited in claim 1 as alleged by the Office.

Furthermore, with respect to the feature "an application program interface (API) that generates a report signal used to identify a buffering state of the markup document; and the report signal is used by the apparatus to verify . . . whether the markup document is being read" recited in claim 1, the Office states as follows:

If the I/O API indicates that a file is not located local to the client (¶68 and 69) and the preloader indicate [*sic*] how much of the resource or chunk it was able to preload before suspending, then the status of the file is that the file is being read.

However, as can be seen from paragraph [0066], lines 5-11, of Jones, the preloader indicates how much of the resource (or chunk of a resource) it was able to preload before suspending at the time it suspends preloading, which is before the file system I/O API generates the "report signal" indicating whether the file or resource has been loaded as described in paragraph [0072], lines 20-24, of Jones. At the time the file system I/O API generates the "report signal" indicating that the partially preloaded file or resource has not been loaded, the partially preloaded file or resource is not being read because the preloader has not yet resumed preloading resources because the preloader is still in the process of determining which resources still need to be preloaded. Furthermore, it is not seen where the "apparatus" in FIG. 12 of Jones ever verifies that a particular resource is being preloaded or read by the preloader.

Accordingly, for at least the foregoing reasons, it is submitted that paragraphs [0066] and [0068] of Jones do not disclose or suggest "an application program interface (API) that generates a report signal used to identify a buffering state of the markup document; and the report signal is used by the apparatus to verify . . . whether the markup document is being read" as recited in claim 1 as alleged by the Office.

Conclusion—Rejection 1

For at least the foregoing reasons, it is respectfully requested that the rejection of claims 1, 3, and 5-7 (i.e., claim 1 discussed above and claims 3 and 5-7 depending directly or indirectly from claim 1) under 35 USC 103(a) as being unpatentable over Kanazawa in view of Jones and Lamkin '021 be withdrawn.

Rejection 2

Claims 10 and 11 have been rejected under 35 USC 103(a) as being unpatentable over Kanazawa in view of Jones, Lamkin '021, and Collart (U.S. Patent Application Publication No. 2002/0088011). This rejection is respectfully traversed.

Although the Office has referred to U.S. Patent Application Publication No. 2002/0088011 as "Collart," it is noted that Collart is the second-listed inventor. The first-listed inventor is Lamkin, and since the standard practice is to refer to a reference by the last name of the first-listed inventor, it is submitted that the Office should have referred to U.S. Patent Application Publication No. 2002/0088011 as "Lamkin '011" (to distinguish it from Lamkin '021), rather than "Collart." However, since the Office has referred to this reference as "Collart," the applicants will also refer to this reference as "Collart."

Although the propriety of this rejection is not conceded, it is submitted that claims 10 and 11 depending from claim 1 are patentable over Kanazawa, Jones, Lamkin '021, and Collart for at least the same reasons discussed above that claim 1 is patentable over Kanazawa, Jones, and Lamkin '021.

For at least the foregoing reasons, it is respectfully requested that the rejection of claims 10 and 11 under 35 USC 103(a) as being unpatentable over Kanazawa in view of Jones, Lamkin '021, and Collart be withdrawn.

Rejection 3

Claims 14, 15, 17-21, 25, and 28-38 have been rejected under 35 USC 103(a) as being unpatentable over Kanazawa in view of Jones. This rejection is respectfully traversed.

Claims 14, 25, 37, and 38

It is submitted that Kanazawa and Jones do not disclose or suggest the following features of independent claim 14:

the apparatus generates the report signal using an application program interface (API); and

the report signal is used by the buffer manager to verify whether the markup document has been successfully preloaded into the buffer, whether the markup document cannot be read due to an error, and whether the markup document is being read,

or the following features of independent claim 25:

generating a report signal used to identify a buffering state of the markup document using an application program interface (API);

using the report signal to verify whether the markup document has been successfully preloaded, whether the markup document cannot be read due to an error, and whether the markup document is being read,

or the following features of dependent claim 37:

generating the report signal using an application program interface (API);

using the report signal to verify whether the markup document has been successfully preloaded, whether the markup document cannot be read due to an error, and whether the markup document is being read,

or the following features of dependent claim 38:

generating the report signal using an application program interface (API);

using the report signal to verify whether the markup document has been successfully preloaded, whether the markup document cannot be read due to an error, and whether the markup document is being read,

for at least the same reasons discussed above that Kanazawa, Jones, and Lamkin '021 do not disclose or suggest the similar features of claim 1.

Claim 32

It is submitted that Kanazawa and Jones do not disclose or suggest the following features of independent claim 32:

buffering the markup document to preload the markup document in response to a fetch signal;

outputting a buffering state of the markup document in response to a report signal;

staging the markup document for decoding in response to a retrieve signal; and

deleting the markup document in response to a discard signal.

The Office considers column 15, lines 34-56, and column 17, lines 31-38, of Kanazawa to disclose "buffering the markup document to preload the markup document in response to a fetch signal" as recited in claim 32; considers paragraphs [0066]-[0068] of Jones to disclose "outputting a buffering state of the markup document in response to a report signal" and "staging the markup document for decoding in response to a retrieve signal" as recited in claim 32; and considers paragraphs [0049] and [0066] of Jones to disclose "deleting the markup document in response to a discard signal" as recited in claim 32. However, the Office did not explain why it would have been obvious to combine Kanazawa and Jones to obtain the combination of features recited in claim 32, such that the Office has not established a *prima facie* case of obviousness with respect to claim 32. Although the Office did provide such an explanation with respect to independent claims 14 and 25, claim 32 recites features that are not recited in claims 14 and 25, such that the Office's explanation with respect to claims 14 and 25 is not applicable to claim 32.

Furthermore, in the event the Office is inclined to provide an explanation of why it would have been obvious to combine Kanazawa and Jones to obtain the combination of features recited in claim 32 in the response to this Amendment After Final Rejection, it is submitted that the Office can only do this in a non-final Office Action because the Office has not met its burden of establishing a *prima facie* case of obviousness in the Final Office Action of August 3, 2010.

The Office states as follows:

Examiner interprets "staging the markup document for decoding" as preloading the document for processing. Therefore, the preloading process disclosed in paragraphs [0066]-[0068] of Jones corresponds to the claimed staging process. In addition, Kanazawa teaches preloading markup documents for decoding (i.e. col. 15 lines 34 – 56; col. 17 line 64 – col. 18 line 23).

However, the Office has apparently overlooked the fact that claim 32 recites both "buffering the markup document to preload the markup document in response to a fetch signal" and "staging the markup document for decoding in response to a retrieve signal." Assuming *arguendo* that the portions of Kanazawa and Jones relied on by the Office may arguably be considered to correspond to "buffering the markup document to preload the markup document in response to a fetch signal" as recited in claim 32, it is not seen where Kanazawa and Jones disclose anything whatsoever that may reasonably be considered to correspond to "staging the markup document for decoding in response to a retrieve signal" as recited in claim 32.

Furthermore, the Office has not identified which elements in the portions of Kanazawa and Jones it has relied on it considers to correspond to the "fetch signal" and the "retrieve signal" recited in claim 32, such that the Office has not established a *prima facie* case of obviousness with respect to these features of claim 32.

The Office considers paragraphs [0049] and [0066] of Jones to disclose "deleting the markup document in response to a discard signal" as recited in claim 32. However, although paragraph [0049] of Jones discloses discarding a pointer to a resource object when all the resources associated with the resource object have been preloaded, it is submitted that paragraph [0049] does not disclose or suggest doing this "in response to a discard signal" as recited in claim 32, or deleting the resource object itself as would be necessary for paragraph [0049] to arguably disclose or suggest "deleting the markup document" as recited in claim 32.

Also, although paragraph [0066] of Jones discloses that the preloader may discard the resource (or chunk of a resource) which it was currently trying to load when the preloader suspends itself from executing, it is submitted that paragraph [0066] does not disclose or suggest that the preloader does this "in response to a discard signal" as recited in claim 32.

The above arguments were also presented on page 17 of the Amendment of May 14, 2010. In response to these arguments, the Office states as follows on page 5 of the Final Office Action of August 3, 2010:

Jones teaches that when the preloader determines that the application has launched, the preloader may suspend itself from executing and discard the resource (or chunk of resource) which it was currently trying to load (¶166). The signal indicating that the application has launched corresponds to the claimed discard signal.

Thus, it appears that the Office is no longer relying on paragraph [0049] of Jones to show the feature "deleting the markup document in response to a discard signal" as recited in claim 32. It is respectfully requested that the Office confirm whether this is correct in the next Office Action, even if that Office Action is an Advisory Action.

However, what is discarded or deleted in paragraph [0066] of Jones relied on by the Office is not a resource or "markup document" as recited in claim 1, but a portion of a resource or "markup document" (or a portion of a chunk of a resource or "markup document").

Claim 35

It is submitted that Kanazawa and Jones do not disclose or suggest the following features of independent claim 35 for at least the same reasons discussed above that Kanazawa and Jones do not disclose or suggest the similar features of claim 32:

generating a fetch signal to preload the markup document;
generating a report signal to determine a buffering state of the markup document;
generating a retrieve signal to stage the markup document for decoding; and
generating a discard signal to delete the markup document,

Conclusion—Rejection 3

For at least the foregoing reasons, it is respectfully requested that the rejection of claims 14, 15, 17-21, 25, and 28-38 (i.e., claims 14, 25, 32, 35, 37, and 38 discussed above and claims 15, 17-21, 28-31, 33, 34, and 36 depending directly or indirectly from claims 14, 25, 32, and 35) under 35 USC 103(a) as being unpatentable over Kanazawa in view of Jones be withdrawn.

Rejection 4

Claims 22 and 23 have been rejected under 35 USC 103(a) as being unpatentable over Kanazawa in view of Jones and Collart. This rejection is respectfully traversed.

Claim 22

Although the propriety of this rejection is not conceded, it is submitted that claim 22 depending from claim 14 is patentable over Kanazawa, Jones, and Collart for at least the same reasons discussed above that claim 14 is patentable over Kanazawa and Jones.

Claim 23

It is submitted that Kanazawa, Jones, and Collart do not disclose or suggest the following features of independent claim 23 for at least the same reasons discussed above that Kanazawa, Jones, and Lamkin '021 do not disclose or suggest the similar features of claim 1:

the ENAV engine generates the report signal using an application program interface (API); and

the report signal is used by the ENAV engine to verify whether the markup document has been successfully preloaded into the ENAV buffer, whether the markup document cannot be read due to an error, and whether the markup document is being read.

Conclusion—Rejection 4

For at least the foregoing reasons, it is respectfully requested that the rejection of claims 22 and 23 under 35 USC 103(a) as being unpatentable over Kanazawa in view of Jones and Collart be withdrawn.

Rejection 5

Claim 24 has been rejected under 35 USC 103(a) as being unpatentable over Kanazawa in view of Jones, Collart, and Silberschatz (Applied Operating System Concepts, First Edition, John Wiley & Sons, New York, 2000, pp, 65-66 and 412-431). This rejection is respectfully traversed.

Although the propriety of the rejection is not conceded, it is submitted that claim 24 depending from claim 23 is patentable over Kanazawa, Jones, Collart, and Silberschatz for at least the same reasons discussed above that claim 23 is patentable over Kanazawa, Jones, and Collart.

For at least the foregoing reasons, it is respectfully requested that the rejection of claim 24 under 35 USC 103(a) as being unpatentable over Kanazawa in view of Jones, Collart, and Silberschatz be withdrawn.

Conclusion

There being no further outstanding objections or rejections, it is submitted that the application is in condition for allowance. An early action to that effect is courteously solicited.

Finally, if there are any formal matters remaining after this response, the Office is requested to telephone the undersigned to attend to these matters.

Please charge any fees under 37 CFR 1.16 and 1.17 that may be required for this paper only to Deposit Account 50-5113 in the name of North Star Intellectual Property Law, PC.

Respectfully submitted,

Date: September 30, 2010

By: /Randall S. Svihla/
Randall S. Svihla
Registration No. 56,273

NSIP Law
1156 15th Street NW, Suite 603
Washington, DC 20005
Tel: (202) 429-0020
Fax: (202) 315-3758
CYP/RSS